DL1 20.02.2024 Gedächtnisprotoll

Dienstag, 20. Februar 2024 14:13

1. Task: Multiple-Choice

1.1 What is a gate in LSTM

- Sigmoid multiplication to real value
- Sigmoid multiplication to positive real value
- ReLU multiplication to real value
- ReLU multiplication to positive real value

1.2 How many weights does an nn.Conv2D(3,20,5) layer have?

- 60
- 300
- 900
- 1500

1.3 Using nn.averagepool(2, stride=4) on a 200×200 input results in

- 0% inputs not used
- 25% inputs not used
- 50% inputs not used
- 75% inputs not used

1.4

-
-
-
-

1.5

-
-
-
-

1.6 The 0/1 loss $1_{\{y_t>0\}}$ is

- Not hard to optimize
- Not outlier robust
- ... to small variations
-

1.7 The perceptron loss can be written as

- $\max(0, -y_t)$
- $\max(0, -y_t)^2$
- $1_{\{y_t>0\}}$
- $1_{\{y_t>\varepsilon\}}$ for some $\varepsilon>0$

1.8 Small local invariances can be addressed by using
- Activation layers
- Pooling layers
- Convolution layers
- Linear layers

---

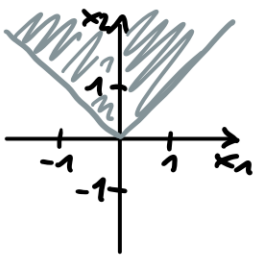
2. Task: Decision boundary

Given: $a_3 = \max(x_1 + x_2)$
$a_4 = \max(x_1 - x_2)$
$y = a_3 + a_4$

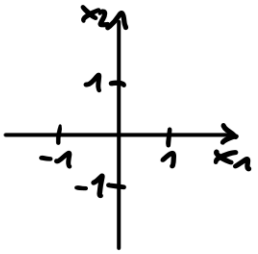
a) Draw the four pieces on which y is linear

Piece 1 (given)

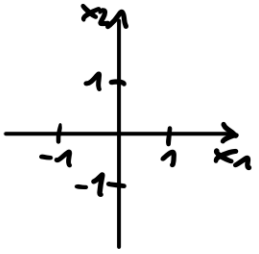
$(x_2 > -x_1) \wedge (x_2 > x_1)$



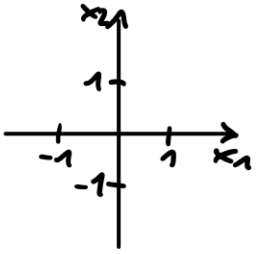
Piece 2



Piece 3



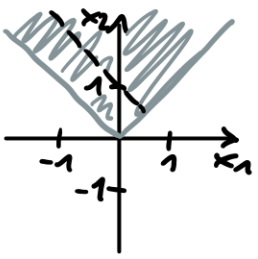
Piece 4



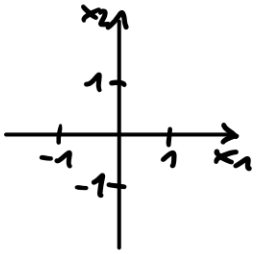
b) Draw the decision boundary on each piece for $y = f(x) = 1$

Piece 1 (given)

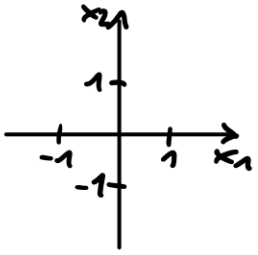
$y = x_1 + x_2$



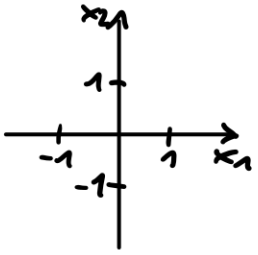
Piece 2



Piece 3



Piece 4



---

3. Task: Gradient, bound and regularization

Given: $z_{ij} = |x_i - c_{ij}|$, $a_j = \exp(-\sum_{i=1}^{d} z_{ij})$, $y = \sum_{j=1}^{h} v_j a_j$

a) Calculate $\frac{\partial y}{\partial x_i}$

b) Show that $\left\|\frac{\partial y}{\partial x}\right\| \leq d\|w\|_1$ Hint: $|\sum_i a_i| \leq \sum_i |a_i|$

c) Explain in 1-2 sentences how this bound helps with choosing the regularization

---

4. Task: RNN

Given: $h_1 = \tanh(x_1^T w + h_0)$
$h_2 = \tanh(x_2^T w + h_1)$
$y = h_2$
$\mathcal{E} = (y-t)^2$

a) Draw the associated NN graph

b) Calculate $\frac{\partial \mathcal{E}}{\partial y}$

c) State the derivative of $\frac{\partial \mathcal{E}}{\partial w}$ using the chain rule

d) Calculate $\frac{\partial \mathcal{E}}{\partial w}$

---

5. Task: Programming

a) Program an adversarial attack on a trained model using $\min_z \{\|x-z\|^2 + \max(0, t \cdot f(z))\}$

```
1  import torch, import...                          x: 1×d
2  def adversarial(x, T, model)                     t: 1×d
3      #Create/modify z                             model(x): 1×1
4
5                                                   use 1000 iterations, stopping not needed
6      optim = torch.nn.SGD([z], 0)
7      #create optimization
8
9
10
11
12
13
14     return z
```

b) How would this code need to be modified to take $x \in \mathbb{R}^{n \times d}$ as input? Use extra loops and provide line numbers!

c) How can the model be made more robust?